한국어 복합명사 분해 오류 탐지 기법

강민규○ 강승식
국민대학교 컴퓨터공학부
pelcious@gmail.com, sskang@kookmin.ac.kr

Error Detection Method for Korean Compound Noun Decomposition

Minkyu Kang○ Seungshik Kang
School of Computer Science, Kookmin University

요 약

복합명사를 분해하는데 있어서 발생하는 분해오류들은 대부분 예외상황들로 취급된다. 전체적으로 차지하는 비중은 크지 않은데 오류 처리를 위해 들어가는 비용이 상대적으로 크기 때문이다. 하지만 분해된 데이터를 색인기나 문서분류기, 기계번역기 등에 실제로 적용해야 할 경우, 분해오류들을 수정해주어야 더 나은 성능을 보일 수 있기 때문에 분해오류를 찾아내고 수정하는 방법을 고안해야 한다. 본 논문에서는 복합명사 분해기에서 추출된 결과를 살펴보고, 주요 분해오류들이 가진 공통적인 특징을 파악하여 분해오류를 발견하는 방법을 생각해보고자 한다.

주제어: 복합명사, 분해오류, bigram, 오류 탐지

1. 서 론

한국어 정보처리 시스템을 구현하는데 있어서 문서 내의 명사와 복합명사를 정확하게 추출하는 문제는 전체적인 시스템의 성능을 향상시키는데 필수적인 요소라고 할 수 있다. 명사의 경우 사전에 등록이 되어있다면 추출이 가능하다. 하지만 복합명사의 경우는 다양한 명사 조합에 의해 이루어지고, 띄어쓰기와 붙여쓰기의 표현이 자유롭기 때문에 모든 경우에 맞게 적용되는 사전을 보유하는 것 보다는 구성명사 단위로 분해하는 방법을 통해 추출을 한다.

한국어 복합명사의 분해는 크게 음절별 주요 분해 패턴에 의해 분해하는 방법[1, 5, 7]과 통계적인 데이터를 통해 분해 지점을 선택하는 방법[2, 3, 4]이 많이 사용된다. 최장일치법을 기준 방법으로 사용[6]하거나 부분적으로 적용[4]하는 방법도 고려되고 있다.

제시된 방법들은 모두 90%내외의 우수한 성능을 보이지만, 미등록어의 존재 유무에 따라 성능의 차이를 보인다. 실험에서는 절대적인 성능평가를 위해 미등록어를 등록한 실험도 포함되어 있지만 실제 상황에서 알고리즘을 사용할 경우에 미등록어를 모두 등록하는 것은 현실적으로 불가능하다. [7]에서는 고유명사와 같은 주요 미등록어 사전을 사용한 복합명사 분해 알고리즘을 제시하고 있으나, 자주 등장하는 주요 고유명사에 대해서만 진행했으므로 모든 미등록어 문제를 해결했다고 보기는 어렵다.

성능에 큰 영향을 미치지 않지만, 1음절어로 인해 발생하는 문제점도 생각해야 한다. [1]에서 언급된 것처럼 전체 패턴에서 1음절 패턴이 갖는 비중이 작고, 알고리즘 전체에 끼치는 영향이 작기 때문에 앞, 뒤 명사와 결합하는 방법을 주로 사용한다. 하지만 적은 양이라고 하더라도 그로 인한 분해 오류가 존재하기 때문에 데이터 적용 이전에 재처리를 고려해야 한다.

2. 관련 연구

복합명사의 분해오류 판단 방법을 고려하기 전에 분해오류를 불러오는 주요 원인에 대하여 살펴보고 그에 대한 해결 방법을 마련하고자 한다.

2.1 미등록 명사의 분해오류

인명, 지명 등의 고유 명사, 외래어의 한국어 표기, 신조어 등이 명사 사전에 포함되어 있지 않은 경우, 복합명사 분해기가 이를 복합명사로 판단하지 않는다. 하지만 해당 미등록어가 다른 명사로 인해 분해가 이루어질 수 있는 경우, 복합명사 분해기는 분해되는 부분을 기준으로 해당 미등록어를 나누어 버리게 된다. '패트리어트'에서 '트리', '프랑크프루트'에서 '프랑'과 같은 경우가 이에 해당한다. '패트리어트'나 '프랑크프루트'는 각각 단일명사로 인식이 되어야 하지만 복합명사 분해기가 '패-트리-어트', '프랑-크프루트'로 분해된 결과를 반환하기 때문에 복합명사처럼 인식되는 문제가 발생하게 된다.

미등록 명사가 구성명사로 사용될 경우에는 또 다른 문제를 불러올 수 있는데 인접한 타 구성명사와 함께 잘못된 분해를 가져오는 경우이다. '박정희대통령'은 '박정희-대통령'으로 분류되어야 옳지만, '박정희'가 미등록어

인 경우, '박정', '희대', '통령'이 모두 명사로 등록된 경우, '박정-희대-통령'이라는 결과가 나타날 수 있다.

미등록어로 인한 문제는 모든 미등록어를 사전에 등록하면 해결되지만, 전체 문서 내에서 고르게 등장할 확률이 극히 낮은 고유 명사를 모두 등록하면 사전의 크기가 매우 커지고, 사전 검색 속도가 떨어지기 때문에 분해 성능의 향상 정도에 비해서 잃는 것이 많아 좋은 방법이 아니다.

2.2 1음절어로 인한 복합명사의 분해오류

접두사와 접미사로 쓰이는 1음절어는 인접 구성명사와 조합되어 두 개의 전혀 다른 구성명사로 나누어질 경우, 분해오류를 가져올 수 있다. '부-위원장'을 '부위-'원장'으로 분해하거나, '정치-경제-학-적'을 '정치-경제-학적'으로 분해하는 경우와 같이 1음절어를 포함한 분해에 있어서 잘못된 분해를 유도하는 구성명사가 사전에 등록되어 있을 경우, 이를 먼저 선택하기 때문에 발생하는 문제이다.

이는 주요 패턴 간의 중의성 문제와는 다른 경우로 볼 수 있는데 '1+', '+1' 패턴의 비중이 적어 무시하고 처리하는 것이 시스템의 성능상 유리하기 때문이다.

2.3 주요 분해오류에 대한 고찰

위에서 살펴본 분해오류의 원인들은 대다수가 두 개의 특징으로 살펴볼 수 있다.

① 분해된 구성 명사중 하나 이상이 명사로 사용되지 않는다.

② ①의 가부와 상관없이 거의 모든 분해오류에 있어서, 중의성 문제에서 보는 것과 마찬가지로 걸맞지 않은 명사간의 조합이 이루어지게 된다.

따라서 복합명사의 분해오류를 판단하기 위해 구성명사의 명사 여부를 판단하거나 연접 구성명사의 공기 정보를 살펴보고, 구성명사간의 연관성을 찾는 방법을 사용한다.

3. 복합명사 분해오류 분석 방법

복합명사 N = $n_1n_2\cdots n_n$이 주어졌을 때, 복합명사 N의 분해에 대한 신뢰도는 모든 구성명사 n_1, n_2, …, n_n의 신뢰도를 살펴보면서 판단한다.

복합명사의 분해는 일부 구성명사에 대한 분해가 옳게 이루어졌다고 하더라도 잘못된 분해가 한 번 이상 나타나게 되면 잘못된 분해로 봐야 한다. 따라서 복합명사 N의 분해 방법에 대한 신뢰도는 모든 구성명사 n_1, n_2, …, n_n의 신뢰도 중 최소값을 사용한다.

$$C(N) = \underset{i}{\overset{n-1}{\arg min}}(I(n_i, n_{i+1}))$$

구성명사의 신뢰도는 연접 구성명사와의 상호 정보량으로 한다. 모든 구성명사의 신뢰도 중 최소값을 복합명사의 신뢰도 C(N)의 값으로 하고, 이 신뢰도가 임계값을 넘을 경우, 복합명사 분해가 잘 되었다고 판단한다.

본 논문에서 사용하는 모든 방법들은 여러 가지 분해 기준 중 최적값을 찾는 방법에 대한 기술이 아니고, 이미 분해가 이루어진 단일 기준에 대해서 옳은가 옳지 않은가를 판단하는 방법에 대한 기술이므로 신뢰도의 확률적 우위보다는 절대적인 수치인 빈도수를 사용하여 가부를 판단한다.

3.1 bigram을 이용한 구성명사 신뢰도 계산

구성명사의 신뢰도를 계산하는데 있어서 n_i가 잘 분해되었는지에 대한 판단은 n_i와 n_{i+1}이 띄어쓰기로 연접한 경우의 상호 정보량을 통해서 살펴본다.

$$I(n_i, n_{i+1}) = \frac{P(n_i, n_{i+1})}{P(n_i)P(n_{i+1})} = \frac{freq(n_i, n_{i+1})N}{freq(n_i)freq(n_{i+1})}$$

복합명사의 구성명사이고, 분해가 잘 이루어진 연속된 두 어절은 복합명사의 표현 방법에 따라 띄어쓰기의 형태로 나타날 가능성이 존재한다. 따라서 띄어쓰기에 대한 공기 빈도수가 일정 수 이상 존재할 경우, 해당 구성명사의 분해 신뢰도는 높다고 판단한다. 예를 들어 '하이네켄'이라는 고유명사를 복합명사로 판단, 구성명사를 분리해버린 경우, 그 구성명사는 '하이', '네켄'이 되는데, '하이-네켄'의 분해가 올바른 형태라면, '하이 네켄'과 같은 띄어쓰기의 형태로 나타날 가능성이 있다. '하이 네켄'은 공기 빈도수가 0이라면, 이 결과를 통해 '하이-네켄'의 분해는 틀렸다고 결론을 내리게 된다. 이와 같이 띄어쓰기에 대한 상호 정보량을 계산, 붙여쓰기 형태의 분해오류에 대한 신뢰도를 구한다.

3.2 수정된 bigram을 이용한 구성명사 신뢰도 계산

bigram을 이용한 구성명사의 신뢰도 계산 방법은 다음과 같은 경우에 있어서 계산이 힘든 단점이 있다.

① 1음절어의 bigram 정보를 찾을 수 없는 경우

② 3음절어(1+2, 2+1)의 bigram 정보를 찾을 수 없는 경우

1음절어의 경우, 접두/접미사가 잘 분해된 경우와 미등록어를 분해하는 과정에서 1음절어로 분해된 경우가 있다. 전자는 '제조-물-책임-법'과 같이 접미사를 인식할 수 있다면 해결이 가능한 형태이고, 후자는 '디-플레이-

션'과 같이 접두/접미사로 볼 수 없는 단어들로 분해된 형태이다.

1음절어 문제를 해결하기 위해 자주 사용되는 접두사, 접미사 사전을 검색, 1음절어에 대한 판단을 따로 진행하고 상호 정보량 계산을 제외시키는 과정을 추가한다. 이 과정은 상호 정보량 계산과 직접 연관이 없으므로 어떤 부분에서도 실행이 가능하지만, bigram 사전을 탐색하는 시간을 줄이기 위해 상호 정보량 계산 이전에 처리하는 것이 좋겠다.

3음절어의 경우는 상호 정보량 계산과 동시에 처리한다. 접두/접미사를 제거하더라도 남은 2음절에 대한 상호 정보량을 계산하는 과정이 필요하기 때문에 별도의 계산을 하지 않고 상호 정보량 계산중에 1음절 처리를 추가한다.

$$c(n_i) \approx \underset{i}{\overset{n-1}{\arg max}}(I(n_i,n_{i+1}), I(n'_i,n_{i+1}), I(n_i,n''_{i+1}), I(n'_i,n''_{i+1}))$$

n′ : 구성명사에서 접두사 제거

n" : 구성명사에서 접미사 제거

구성명사 n_i 또는 n_{i+1}가 3음절 이상일 경우, 접두사와 접미사를 분리하고 남은 단어들에 대한 상호 정보량을 구해 그 최대치(가장 많은 빈도수)를 n_i의 신뢰도 값으로 한다.

1음절어와 3음절어의 판단을 위한 접두사와 접미사 사전은 따로 분류해서 사용한다. 3음절어의 처리 과정 중, 잘못된 분해로 인해 접두사 혹은 접미사가 이웃한 구성명사 쪽으로 분해가 되는 경우가 있기 때문이다. '대통령-직인수-위원회'와 같은 경우, 접미사 '직'이 다음 구성명사 쪽으로 분해가 되었기 때문에 상호 정보량 계산 시에는 접두사로 분해하는 과정이 진행된다. 접두사와 접미사 사전을 통합하는 경우, 이와 같은 상황에서 잘못된 판단이 나올 수 있으므로 두 사전을 나눠서 사용한다.

3.3 unigram을 이용한 구성명사 신뢰도 계산

bigram과 수정된 bigram 방법을 사용한 구성명사의 신뢰도가 임계값을 넘지 못한 경우, 이를 바로 분해오류로 처리할 것인가에 대한 문제가 존재한다. 잘 분해된 두 구성명사의 띄어쓰기 공기 빈도수가 모두 존재한다면 bigram 방법에 의해 모든 분해오류가 분류되겠지만, 1음절어, 3음절어의 경우를 제외하고도 잘 분해된 구성명사의 bigram 공기 빈도수가 임계값을 넘지 못하는 상황이 존재할 수 있다. 이와 같은 경우에 대해서 bigram 공기 정보가 아닌 unigram 단일 정보를 통해 판단한다.

unigram을 이용한 방법의 경우, bigram을 이용한 분해오류 판단 이후에 진행되는 과정이므로 bigram 방법에서 분해오류로 판단된 복합명사에 대해서만 분해오류가 맞는지 다시 판단한다. unigram에 의한 분해오류 판단은 모든 구성명사 n_i 각각의 출현 빈도수($freq(n_i)$)와 현재 위치에서의 출현 빈도수($freq_{loc}(n_i)$)가 임계값 τ_n, τ_{loc} 이상인지를 살펴보고 결정한다.

$$C_{uni}(N) = \begin{cases} true, & \forall ic_n(n_i) > \tau_n \\ & \forall ic_{loc}(n_i) > \tau_{loc} \\ false, & others \end{cases}$$

$$c_n(n_i) = P_n(n_i) = \frac{freq_n(n_i)}{\sum freq_n(n_i)} \approx freq_n(n_i)$$

$$c_{loc}(n_i) = P_{loc}(n_i) = \begin{cases} \frac{freq_{begin}(n_i)}{freq(n_i)} \approx freq_{begin}(n_i), \text{if}(i=1) \\ \frac{freq_{end}(n_i)}{freq(n_i)} \approx freq_{end}(n_i), \text{if}(i=n) \\ \frac{freq_{mid}(n_i)}{freq(n_i)} \approx freq_{mid}(n_i), others \end{cases}$$

4. 복합명사 분해오류 분석 알고리즘

위에서 제시된 bigram과 unigram 정보를 이용한 복합명사의 분해오류 추출 방법을 알고리즘 기술 언어로 작성한 내용은 다음과 같다.

```
algorithm compoundErrorDetection(compound)
begin
    for each component in compound
        if(!is1Syllable(component[i]) then
            cn[i] = bigramDetection(component[i])
        end if
    end for
    if (min(cn) < threshold) then
        return unigramDetection(compound)
    else
        return true
    end if
end
```

여기서 is1Syllable은 주어진 구성명사 component[i]가 접두사, 접미사 사전에 등록된 1음절어인지 아닌지 확인하여 참, 거짓을 반환하는 함수이다. 등록되지 않은 1음절 구성명사, 2음절 이상의 구성명사는 거짓을 반환한다. is1Syllable이 거짓을 반환하면 접두/접미사가 아닌 1음절어, 혹은 2음절 이상의 구성명사라고 판단을 내리고, 수정된 bigram에 의한 계산 방법을 적용한다.

bigramDetection은 수정된 bigram 방법에 의해 두 구성명사 component[i], component[i+1]의 상호 정보량을 계산한다. 복합명사의 마지막 구성명사는 상호 정보를 계산하지 않고 지나간다. bigramDetection을 사용하여 계산한 각 component[i]의 상호 정보량 cn[i]의 최소값이 임계값을 넘지 못한 경우, unigramDetection을 사용하여, 오류 확인을 하게 된다. 이미 한번의 오류 확인을 한 상황이므로, bigram 계산법에 의해 오류로 예측된 복합명사에 대해서만 계산을 진행하고, 모든 구성명사의 특정 위치 등장 확률이 임계값을 초과하는지 확인하여 참, 거짓을 판단한다.

5. 실험 및 결과

제시된 알고리즘에 대한 실험은 세종계획 2007의 문어원시 말뭉치 6700만 어절에서 추출된 데이터를 사용해서 진행한다. 데이터에 대한 정보는 표 1과 같다.

표 1 세종말뭉치 정보

전체 어절	6,700만 어절
구성명사	66만 단어
bigram 공기 정보	205만 조합

위의 내용 중 구성명사는 붙여 쓴 복합명사와 미등록어가 포함된 결과로, 본 실험에서 띄어쓰기의 빈도를 기준으로 붙여쓰기의 가부를 추측하기 위해 띄어쓰기 정보만을 수집하여 빈도를 계산하였다.

실험은 세종말뭉치에서 추출된 임의의 붙여 쓴 복합명사 1,100 단어를 대상으로 이루어진다. 주어진 복합명사들은 옳게 분해된 1,000 단어와 잘못 분해된 100단어로 구성되어 있으며, 두 단어 군 각각의 가부를 판단하기 위해 ROC(receiver operating characteristic) 방법에 의해 실험 내용을 분석하도록 한다.

표 2 ROC 비용 행렬

	P	N
P′	True Positive(TP)	False Positive(FP)
N′	False Negative(FN)	True Negative(TN)

P 옳게 분해된 실험군
N 잘못 분해된 실험군
P′ 실험에 의해 옳게 분해된 것으로 예측
N′ 실험에 의해 잘못 분해된 것으로 예측

ROC는 분류기의 정확도에만 의존하지 않고, 가부 판단에 의한 비용까지 고려한 방법으로 여러 분류기의 성능 비교에 사용되는 방법[8]이지만, 본 실험에서는 단순히 잘못된 판단에 따른 비용을 확인하는 기준으로 사용하고자 한다.

분해오류 추출에 있어서 세가지 수치를 살펴볼 필요가 있는데, NPV(negative predictive value)는 TN/(TN+FN)으로 negative에 대한 precision, TNR(true negative rate)는 TN/N으로 negative에 대한 recall, ACC는 accuracy로 TP, TN을 포괄한 성공률이다.[9] 분해오류 판단에 대한 성능은 TNR을 살펴보고, 전체적인 판단에 의한 재분해 비용은 NPV와 ACC를 통해 살펴보도록 한다.

첫 번째 실험은 bigram 공기 빈도를 사용한 방법에 대해서 공기 빈도 임계값을 3으로 하고 실험을 진행 하였다. 그 결과는 표 3과 같다.

표 3 bigram 공기 빈도에 따른 분해오류 판단

	P	N
P′	703	2
N′	297	98

NPV = TN/(TN+FN) = 0.23
TNR = TN/N = 0.98
ACC = (TP+TN)/(P+N) = 801/1,100 = 0.73

bigram 공기 빈도는 분해오류인 복합명사를 추출하기 위해 제안한 방법인 만큼 분해오류로 나타나는 문서에서는 높은 성공률을 보인다. 하지만 옳게 분해된 복합명사이면서도 공기 빈도가 없는 경우를 고려하지 않기 때문에 NPV값과 ACC값이 낮게 나왔다. 실제 옳은 복합명사와 잘못된 복합명사의 비율이 10:1인 것을 감안하더라도 잘못된 복합명사의 재분해에 대한 비용이 실제 사용되어야 할 비용의 4배-NPV가 1/4이므로-가 되는 것은 바람직하지 못한 결과이다.

재분해 비용을 고려했을 때, NPV혹은 ACC가 높아지는 방향으로 접근해야 하고 그에 따라 위에서 언급한 unigram, 수정된 bigram을 적용하여 다시 실험을 진행한다. 각각의 결과는 표 4, 표 5와 같다.

표 4 unigram 방법을 추가한 분해오류 판단

	P	N
P′	885	16
N′	115	84

NPV = TN/(TN+FN) = 0.42
TNR = TN/N = 0.84
ACC = (TP+TN)/(P+N) = 969/1,100 = 0.88

잘못 분해된 복합 명사가 내부적으로는 모두 명사로 인식되는 경우가 있기 때문에 unigram 방법을 적용했을 시, TNR이 떨어지게 된다. 더욱이 unigram 방법에서 1

음절어를 명사로 인식하도록 했기 때문에 FP의 값이 크게 늘어나게 된다. NPV가 올라갔기 때문에 재분해 비용은 줄어들었으나 잘못된 분해에 대한 판단 또한 고려를 해야 하기 때문에 unigram 방법을 전적으로 신뢰할 수가 없다. 따라서 수정된 bigram을 통한 1음절어 처리를 통해서 이를 해결할 수 있는지 살펴본다.

표 5 수정된 bigram 방법을 사용한 분해오류 판단

	P	N
P′	926	9
N′	74	91

NPV = TN/(TN+FN) = 0.51
TNR = TN/N = 0.91
ACC = (TP+TN)/(P+N) = 1,017/1,100 = 0.92

수정된 bigram 방법을 unigram과 같이 사용할 경우 1음절어 미처리로 인해 발생하는 문제와 옳은 분해를 분해오류로 인식하는 문제를 수정할 수 있다. 재분해 비용 또한 실제 사용되어야 할 비용의 2배 이하로 떨어졌기 때문에 허용 가능한 범위로 볼 수 있다.

6. 결론

1음절어 처리를 포함한 bigram 공기 빈도수를 통해서 복합명사의 분해오류를 찾아내는 실험은 비교적 성공적으로 볼 수 있다. 패턴에 따른 분해를 사용하는 복합명사 분해기와 같이 사용할 경우, 전체적인 성공률을 올릴 수 있을 것이라고 예상한다. 하지만 1음절어 처리가 부족하여 분해 오류를 탐지하지 못한 경우들이 존재하는데, '수석-부위-원장', '복합-선거-구제' 등과 같이 2-1-2-1, 2-2-1-1 패턴이 2-2-2 패턴으로 분해된 경우에 대해서는 모든 2음절어를 두 개의 1음절어로 분해해서 살펴볼 수 없기 때문에 문제로 남겨지게 된다. 비슷한 문제로 1음절+조사의 경우도 2음절 패턴으로 인식되는 경우가 존재하는데, 이러한 문제들에 대한 고찰을 계속적으로 진행하여야 할 것으로 보인다.

참고 문헌

[1] 강승식, "한국어 복합명사 분해 알고리즘", 정보과학회논문지(B), 25권, 1호, pp.172-182, 1998

[2] 윤보현, 조민정, 임해창, "통계 정보와 선호 규칙을 이용한 한국어 복합명사의 분해", 정보과학회논문지(B), 24권, 8호, pp.900-909, 1997

[3] 윤준태, 정의석, 송만석, "명사간 어휘 정보를 이용한 한국어 복합 명사 분석", 정보과학회논문지(B) 25권 11호, pp.1716-1725, 1998

[4] 심광섭, "합성된 상호 정보를 이용한 복합 명사 분리", 정보과학회논문지(B), 24권 11호, pp.1307-1317, 1997

[5] 최재혁, "음절수에 따른 한국어 복합 명사 분리 방안", 제8회 한글 및 한국어 정보처리 학술발표논문집, pp.262-267, 1996

[6] 이현민, 박혁로, "복합명사의 역방향 분해 알고리즘", 제12회 한글 및 한국어 정보처리 학술발표논문집, pp.56-59, 2000

[7] 김응균, 서영훈, "미등록어 처리가 강화된 복합명사 분해", 제15회 한글 및 한국어 정보처리 학술발표논문집, pp.40-46, 2003

[8] 김지현, "ROC and Cost Graphs for General Cost Matrix Where Correct Classifications Incur Non-zero Costs", 한국통계학회논문집, 11권, 1호, pp.21-30, 2004

[9] T.Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers", Technical report hpl-2003-4, 2003